

## **APPENDIX A: ALECALL PROGRAM**

This appendix contains both general and detailed software charts of the ALECALL program. The charts describe the logic flow of the program and of key modules, the modular structure of the program, and the data flow. The ALECALL overview logic flowchart explains the ALECALL program execution (Figure A-1). The program is executed with command line arguments. The switches that invoke the various modes of operation are explained in the help menu which may be accessed by typing “alecall” without any other command line arguments. No switches are used to run the program in normal mode; switches are used to change the expected input format, output file, or add debugging, or chimes. The name(s) of the input protocol files to be made into sound files must be designated on the command line for the program to run. Wildcard designations for the file names, such as “\*” or “?”, may be used. Typing “alecall filename.pro” invokes the program in normal mode. The program initializes its variables and structures, parses the command line to set the mode of operation, and opens the first protocol file on the command line. It reads each line in the protocol file, ignoring comments. The noise, fading, or multipath (NFM) input parameters are stored until an initialize command is read, at which point the degraded conditions are set and the filters are made. Degraded conditions can be turned on and off anywhere in the protocol file. ALECALL modulates the ALE word into the 49 ALE tones. The ASCII octal representation of the 49 tones is recorded in a tone file named filename.ton and the tones are converted to 16-bit, 44.1-kHz sampled cosine waves. These waves are written to the sound file after any specified NFM conditions are added to the signal. The process is repeated for the ALE word as many times as designated in the line of the protocol file. When all the repetitions are written to a file, the next line is read and processed, until the end of the file is reached. The file handling procedure is repeated, reading each protocol file specified on the command line, until all of the protocol files have been converted to tone and sound files.

The ALECALL structure chart (Figure A-2) shows the different modules of the program and the calling hierarchy (which procedures are called and by whom). Main parses the command line and calls make\_it, which does most of the work of the program, to process each input file. The last overview chart is the high level data flow diagram (Figure A-3). Flowcharts for the most important program modules follow. There are three logic flowcharts: the main module (Figure A-4), the NFM module (Figure A-5), and the make\_it module (Figure A-6). The main module logic flowchart shows the flow of the main body of the program. The NFM module explains the procedure for turning NFM conditions on and off. It also shows when new conditions are saved or used. The make\_it flowchart describes how each line of the input file is parsed and converted into a sound file representing an ALE call.

The ALECALL program has several modes of operation. It can directly convert a tone file into a sound file. This is the way the two calibration tracks on the Clean Tone CD-01a were created: the second track 1,000-Hz tone and the third track stair-step of all 8 ALE tones. The tone file must be in the exact format as that of a tone file produced by the program. The process is a truncated version of the normal execution, beginning at the point where the ALE tones are converted to cosine waves; it is invoked with a command line switch. The program reads the input tone file and converts the octal representations of the ALE tones into the 16-bit, 44.1-kHz sampled cosine waves and writes

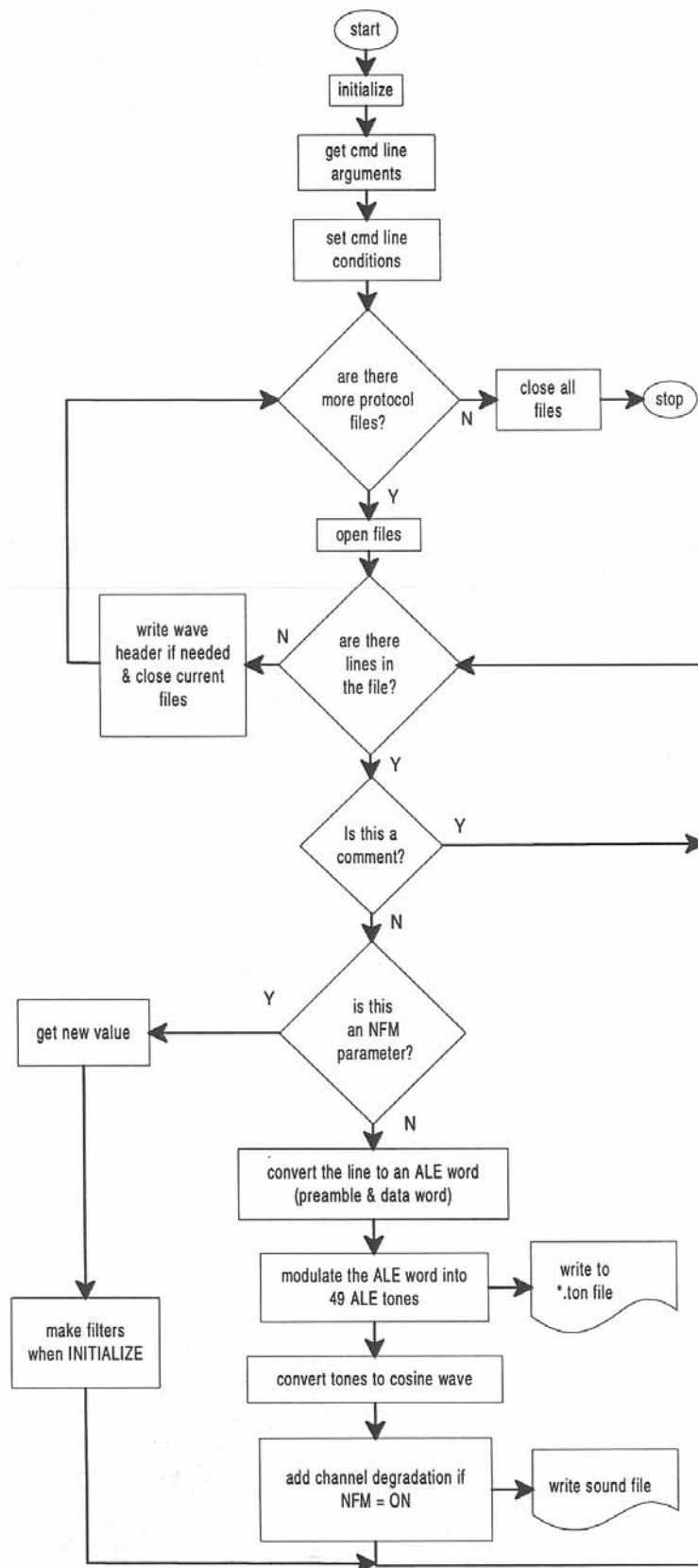
the samples to the sound file. Only clean tone files may be created since the channel degradation conditions are specified in the protocol file. Because the first part of the ALE conversion process (building the tone file) is already accomplished before the program begins and channel degradation is not added, this mode is the fastest method of producing the sound files.

The ALECALL program can also be run to produce only tone files. In this mode the input protocol files are quickly converted to tone files. Although NFM parameters can be specified in the protocol file, they will be ignored since the tone file contains no NFM information. Another operating mode, the verbose mode, may be run concurrently with any of the others. This mode provides additional information on the computer screen to assist and inform the user of the program operations.

Four utility programs are included with the ALECALL program. These programs facilitate the use of the ALECALL software by either encoding difficult features of ALE or providing information on playing the sound files. ALETIME uses the input protocol files to compute the running times of the sound files. The protocol files to be computed are specified on the command line when the program is invoked. The program writes, to the computer screen, the running times of each individual file and the total running time.

The other three utility programs, C-AMD, C-DTM, and C-LQA, assist the user in preparing the protocol file with the following ALE call features: AMD messaging, DTM messaging, and LQA transmissions and requests. Protocol files produced by the utility programs can be read by ALECALL. C-AMD encodes the AMD message specified on the command line and writes it to a protocol file. The command line parameters include the call signs of the sender and the receiver, and either the message in single quotes or the name of the file containing the message. C-DTM is invoked in exactly the same manner with the addition of the kd4 [1] specification: "1" for acknowledgement request or "0" for no acknowledgement request. C-LQA also uses the call signs of the sender and receiver as input parameters. In addition, a control bit, the SINAD, BER, and debug mode may be specified if desired. If an LQA response is chosen, the SINAD and BER are not included. The help screens for these utility programs are in Appendix F, CD-INFO Information for Users of the ALE Clean Tones Compact Disc 01a or may be invoked by typing the program name with no other command line parameters.

Figure A-1. ALECALL overview logic flowchart for normal execution.



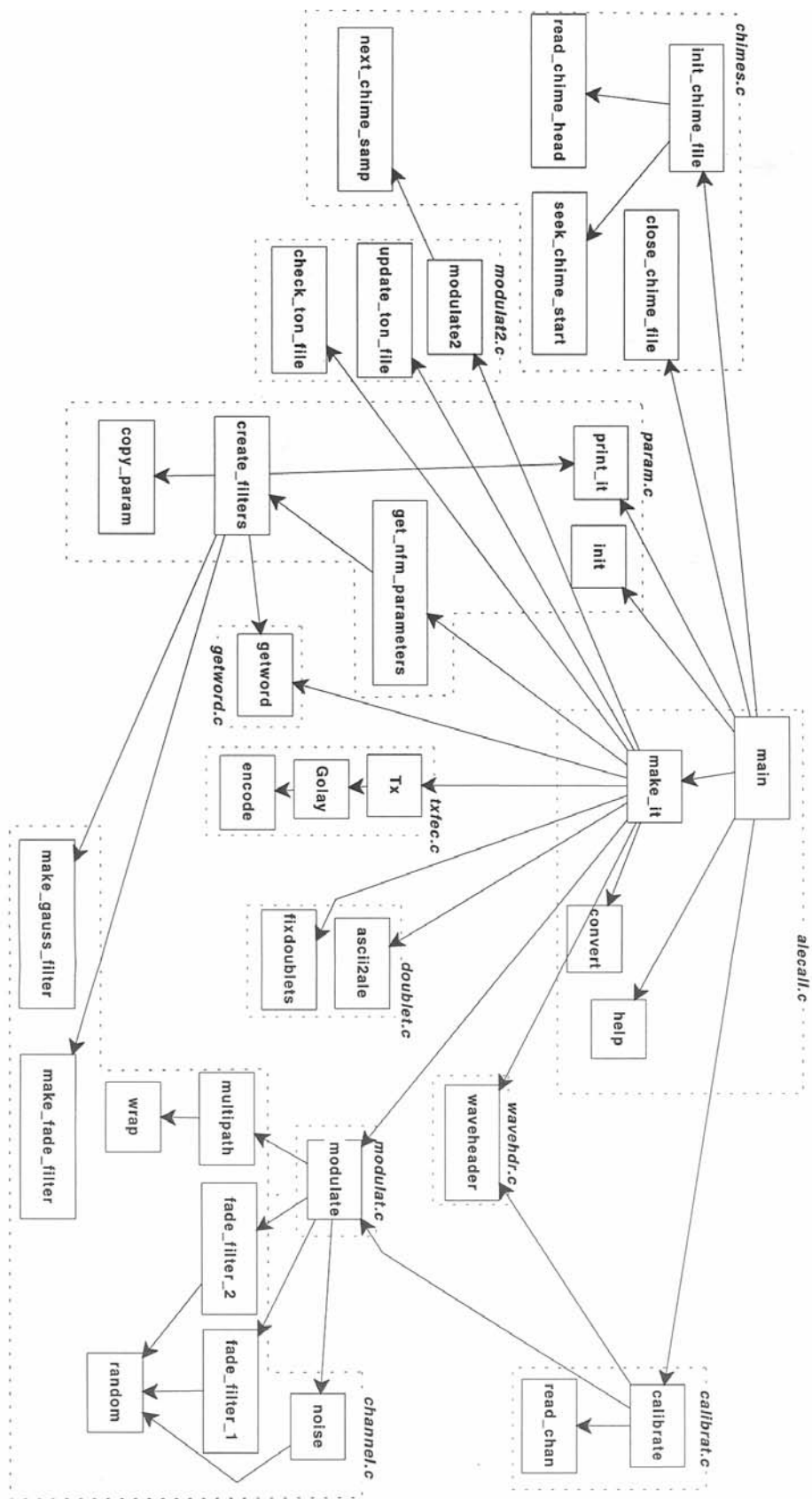


Figure A-2. ALECALL structure chart.

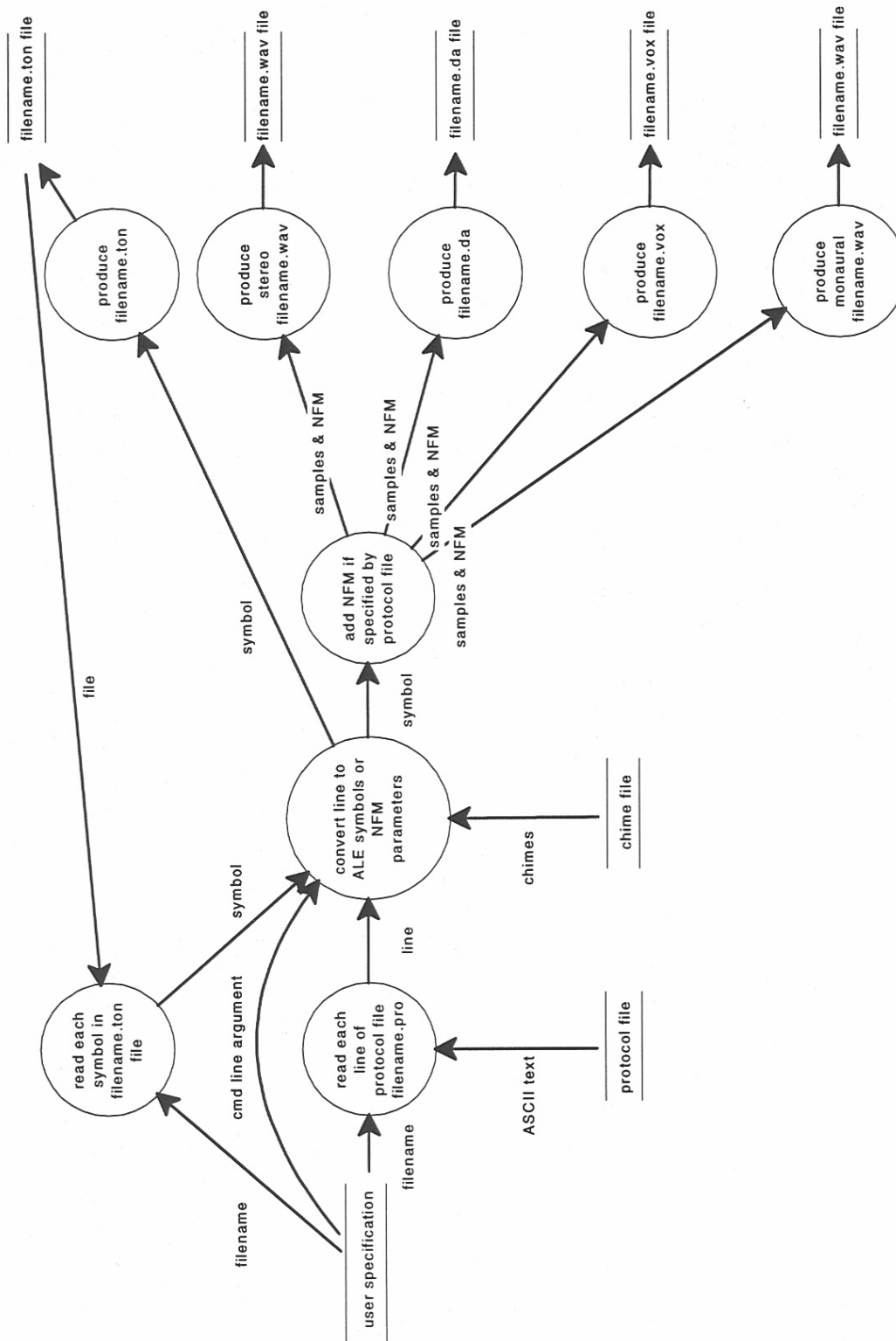


Figure A-3. ALECALL high level data flow diagram.

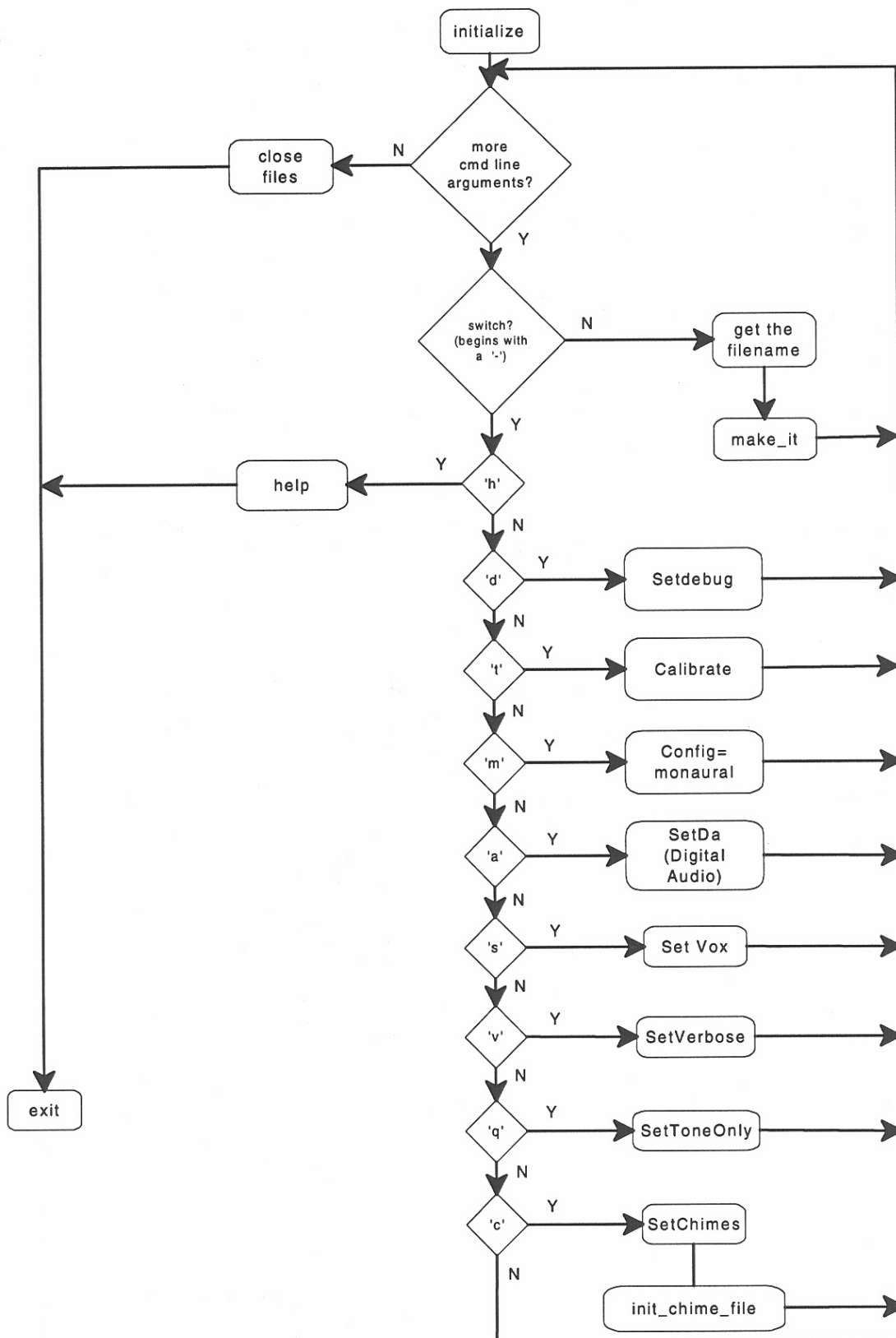


Figure A-4. ALECALL logic flowchart - Main module.

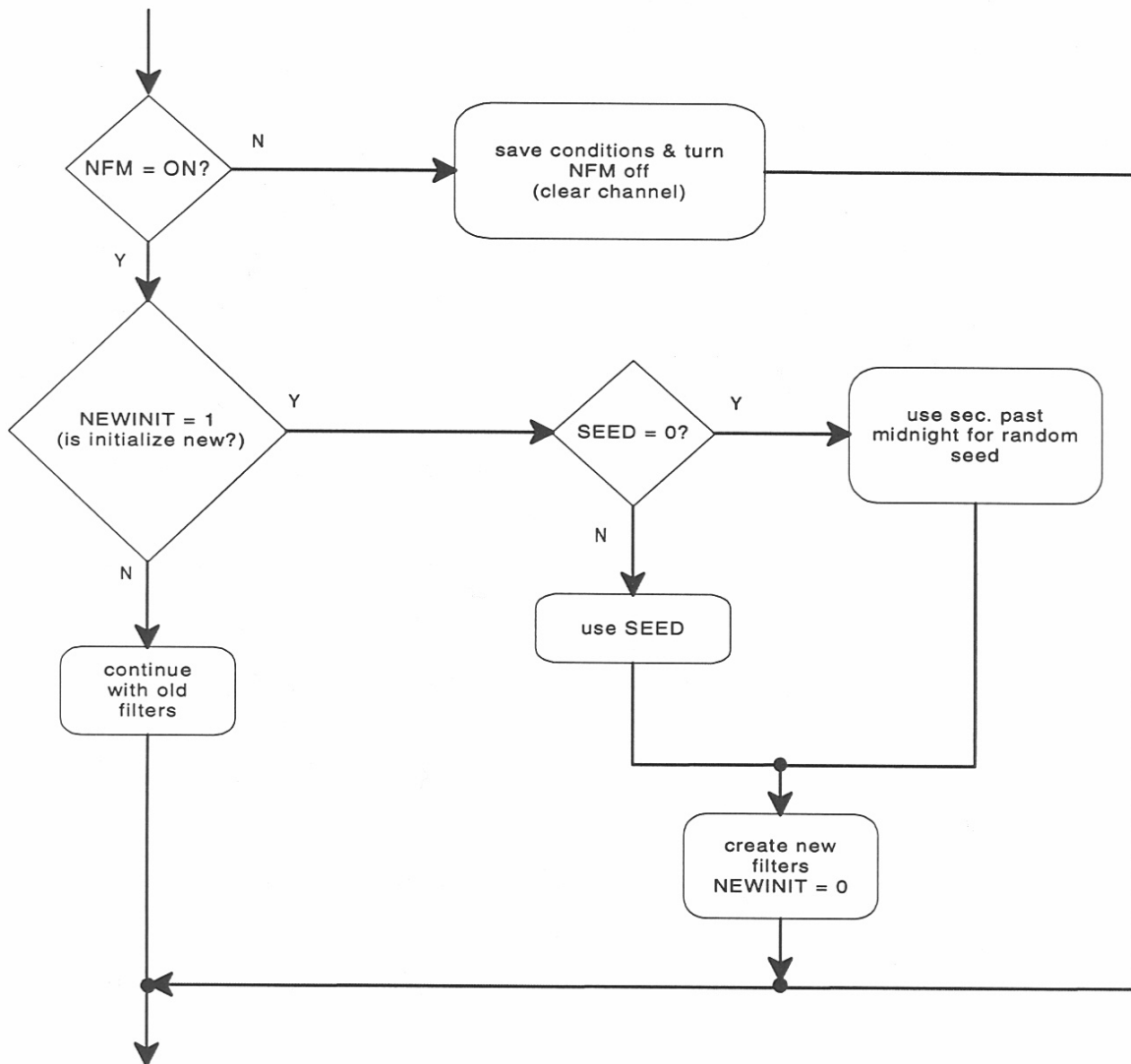


Figure A-5. ALECALL logic flowchart - NFM module.



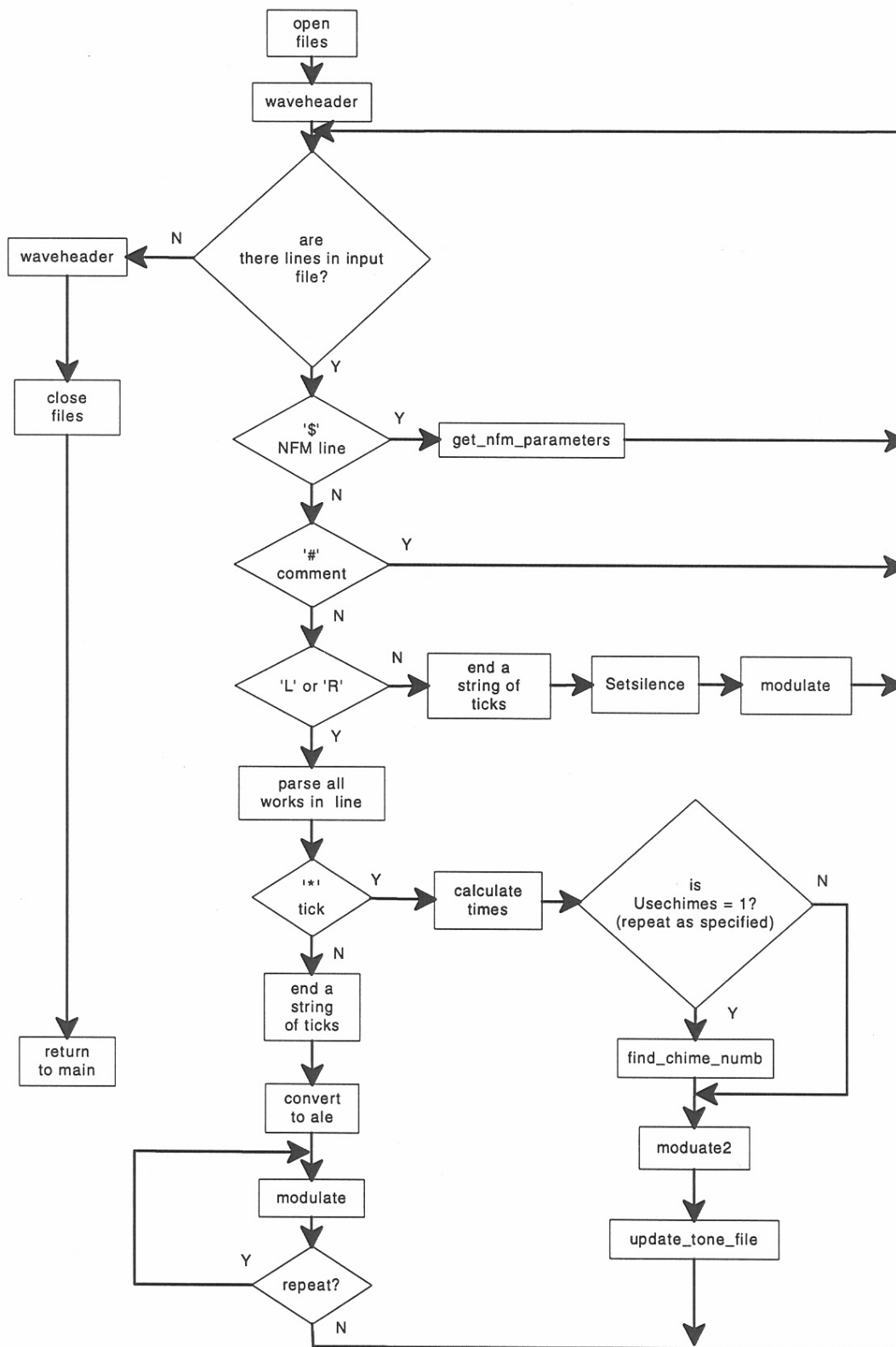


Figure A-6. ALECALL logic flowchart - Make\_it module.

